

How to create, test and deploy XinFin Smart Contracts?

 leewayhertz.com/create-xinfin-smart-contracts



Since XinFin was launched in the blockchain space in 2019, it has brought new decentralized applications. However, XinFin is attributed to the EVM compatible implementation of smart contracts.

XinFin blockchain platform works on the functionality of smart contracts. It is a very common myth that people believe smart contracts are a new concept and are emerged with the Ethereum blockchain platform. But the concept of smart contracts was introduced in 1996 when computer scientist Nick Szabo discovered the term “smart contracts” and defined them as:

“a set of promises, specified in digital form, including protocols within which the parties perform on these promises.”

Before we go in-depth into creating and deploying the XinFin smart contract in this article, it is essential first to understand the XinFin platform and how it works.

XinFin Blockchain Platform

XinFin is one of the leading blockchain platforms that execute arbitrary code to help you run any program on XinFin XDPoS Network. It is an enterprise-ready hybrid blockchain for the global trade and finance market.

XinFin platform is designed to enable tokenization, digitization and quick settlement of trade transactions. Reducing reliance on complex FX infrastructures and increasing efficiency allows increased flexibility in liquidity management for financial institutes.

Using XinFin smart contracts, you can:

- **Create cryptocurrencies**

XinFin XDPoS network allows you to create a tradable token that can be used as a currency. Tokens minted with the XinFin platform use a standard coin API compatible with any XinFin blockchain wallet.

- **Develop dApps**

Developers can build secure and fault-tolerant dApps on the XinFin platform that bring transparency and remove intermediaries from the ecosystem.

- **Raise funds**

You can also use XinFin smart contracts for fundraising. For example, you can create a smart contract and set a deadline. If you fail to achieve the goal,

- **Build virtual organizations**

You can create a smart contract to develop a blockchain-based organization. By adding people to your organization, you can enable voting rules. The smart contract executes automatically when members of the organization vote and reach the needed number of votes.

How does the XinFin blockchain platform execute smart contracts?

To understand how XinFin executes smart contracts, it is essential to know about the execution environment first.

XinFin is an Ethereum Virtual Machine (EVM) compatible blockchain platform. EVM serves as a runtime environment for smart contracts written on the XinFin network. As the name suggests, it is not physical; but a virtual machine. The functionality of EVM is limited as compared to that of the virtual machine; for example, it cannot generate random numbers or make delayed calls on the internet.

Coding programs in assembly language does not make any sense, so, XinFin needed a programming language for EVM. Solidity is a programming language used on XinFin.

Now, let's have a walkthrough of the creation and deployment of XinFin Smart Contracts using Solidity.

A step-by-step guide to creating and deploying XinFin smart contracts

Step1: Install prerequisites

- The network for running and deploying contracts
- Wallet CLI: Wallet-CLI is the XDC Network's command line wallet. Developers can use wallet-cli to deploy contracts on the mainnet and execute other operations.
- XinPay Chrome Extension

XinPay acts as both a wallet and XinFin browser. Using XinPay, you can interact with dApps and smart contracts without installing any software or blockchain. You just need to add XinPay Chrome Extension to your browser, create a wallet and submit XDC.

Before writing smart contracts, you must have some amount of XDC tokens in your XinPay wallet.

Step2: Create a wallet with XinPay

Install XinPay in Google Chrome browser and enable it. After installing it, click on XinPay's icon on the top right of the browser.

Enter the password you want to set and click on the "Create" button. It will then send you a secret backup code that can be used for backing up and restoring the account. Do not share it with anyone and keep it safe somewhere.

You should keep it safe either on an external encrypted hard drive or on a piece of paper. The next step is to ensure that you are in the XinFin Mainnet.



XinPay

Encrypt your new DEN ?

New Password (min 8 chars)

Confirm Password

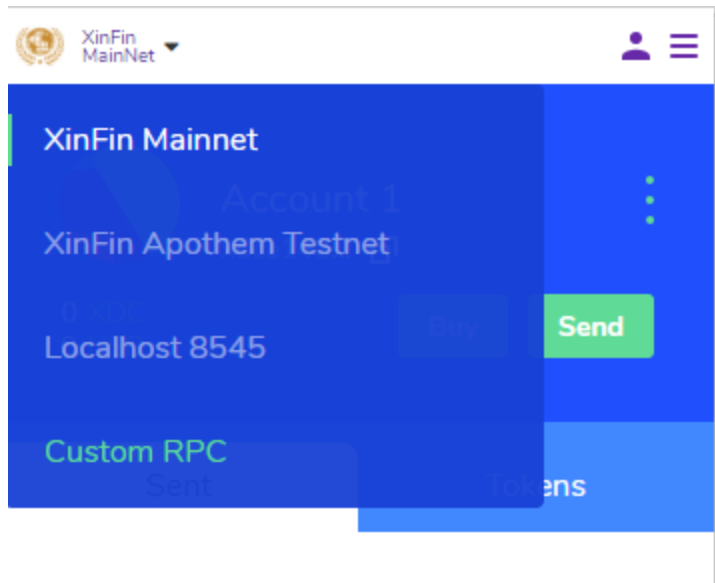
Create

[Import Existing DEN](#)

Step3: Select the network

You need to select the network for creating XinFin smart contracts:

In this article, we will create smart contracts using the Apothem test network.



Step4: Add dummy XDC to your wallet

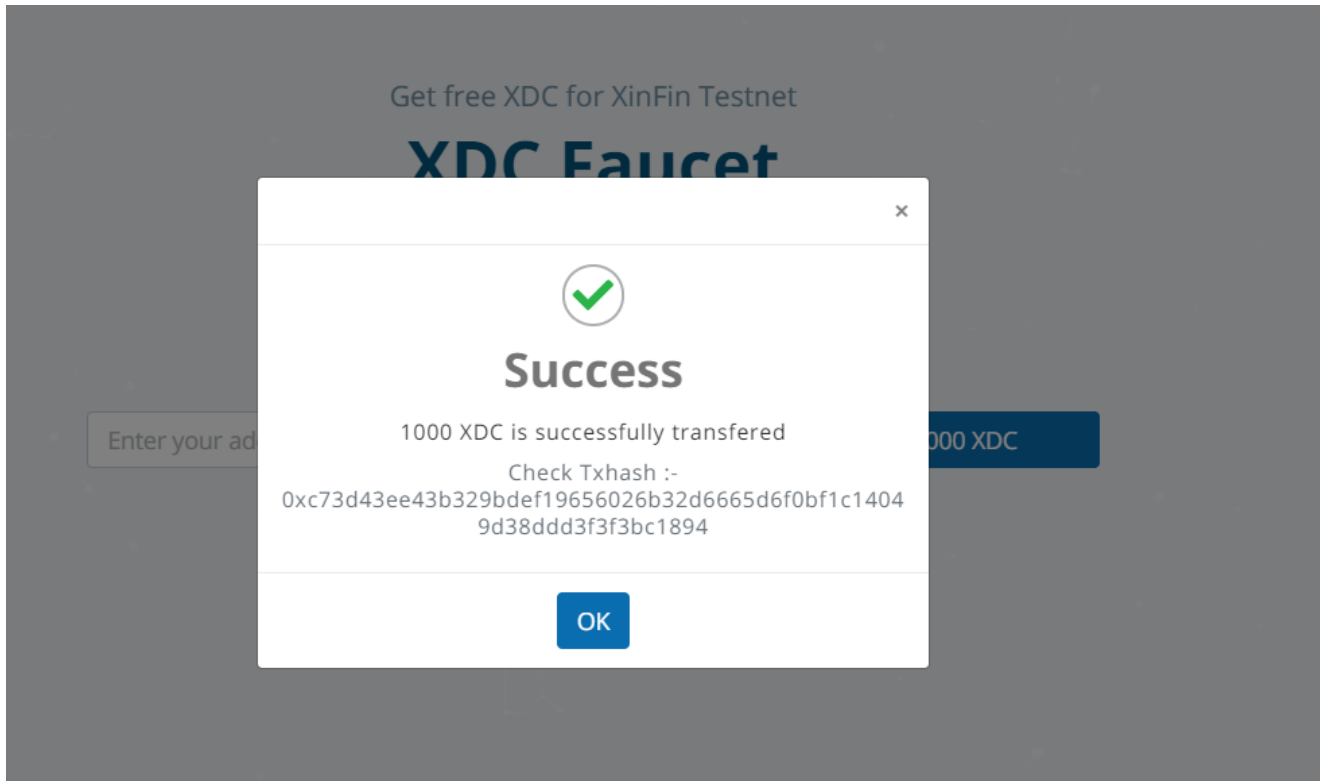
You need to have some dummy XDC in your XinPay wallet to test the smart contract. For example, if you need to test a contract using a test network, select that network and you will see 0 XDC as the initial balance in your account.

To add dummy XDC, visit the faucet and enter your XDC address and you will get 1000 XDC for testing smart contracts.



© 2020 XinFin. All Rights Reserved.

After receiving 1000 XDC, you will view pop up like this:



Once you have dummy XDC in the XinPay wallet, you can start creating smart contracts and coding on the XinFin Remix Browser IDE in Solidity programming language.

Step5: Use XinFin Remix Editor to write smart contracts

We are using XinFin Remix Browser IDE to write smart contracts in Solidity language. This IDE is used for writing small-sized contracts. Features of XinFin Remix Browser IDE include:

- Syntax and error highlighting
- Integrated debugger
- Deploy to XinPay
- Static analysis
- Integrated testing and deployment environment
- Functions with injected web3 objects
- Warnings, including gas cost, checks for overlapping variable names, unsafe code

Step6: Creating a .sol extension file

Open XinFin Remix Browser and click on plus icon on the top left side to create a .sol extension file.

Select a compiler version from XinFin Remix for compiling the Solidity XinFin smart contract code. XRC20.sol is a standard template for creating XRC20 tokens.

```

pragma solidity ^0.4.0;

import "./XRC20.sol";

contract myToken is XRC20{

mapping(address =>uint256) public amount;

uint256 totalAmount;

string tokenName;

string tokenSymbol;

uint256 decimal;

constructor() public{

totalAmount = 10000 * 10**18;

amount[msg.sender]=totalAmount;

tokenName="Mytoken";

tokenSymbol="Mytoken";

decimal=18;

}

function totalSupply() public view returns(uint256){

return totalAmount;

}

function balanceOf(address to_who) public view

returns(uint256){

return amount[to_who];}

function transfer(address to_a,uint256 _value) public

returns(bool){

require(_value<=amount[msg.sender]);

amount[msg.sender]=amount[msg.sender]-_value;

```

```
amount[to_a]=amount[to_a]+_value;  
  
return true;  
  
}  
  
}
```

Step7: Deploying the contract

Deploy smart contract at XinFin Apothem Test Network by clicking the “deploy” button at the left side of the XinFin Remix IDE Window.

You need to wait until the transaction is processed. After committing the transaction, the address of smart contracts can be seen on the left side of the Remix Window.

Firstly, all XRC20 tokens will be stored in the user’s wallet who is deploying the contract.

Go to the XinPay window to check tokens in the wallet. Click “add tokens”, enter the address of the smart contract and click Ok. You would see the number of tokens here.

Now, let’s discuss the steps for testing XinFin smart contracts.

Here are the steps to test the XinFin Smart Contract

- Ensure to run all methods of smart contracts, including totalSupply, balanceOf and transfer. These methods can be found at the left-hand side of the XinFin Remix Window and run all methods from there.
- Make sure if tokens can be transferred to another XinFin wallet address and check the balance of that address by calling balanceOf method.
- Switch to the main XinFin network at XinPay to make smart contracts live.
- Add real XDC to your account.
- Now, you need to deploy the smart contract again using XinFin Remix.
- Once the smart contract is deployed successfully, visit <https://explorer.xinfin.network> and find your smart contract address. Then, select your smart contract.
- Click “verify contract” to verify your smart contract.
- Copy the smart contract code and paste it at XinFin explorer. Select the compiler version that you chose at XinFin remix for compiling the code.
- Also, make sure to check “optimization” to yes if you selected optimization at XinFin Remix. Otherwise, check “optimization” to No.
- Click verify and it will take a few minutes and smart contracts will be live if no issue occurs.

We have used XinFin Remix to create and deploy smart contracts. However, there are different functions and tools that we can explore to write, test and deploy XinFin smart contracts.

If you are looking for smart contracts development services, we can help you. We have a team of experienced blockchain developers who can create and deploy smart contracts on various blockchain platforms, including XinFin, Tezos, Ethereum and more.